## WHAT IS CLAIMED IS:

1.

1

2	an in	terface to receive a packet;
3	a cache to store context data for the packet; and	
4	a pro	cessing engine to process the packet using context data in the cache.
1	2.	The system of claim 1, further comprising:
2	a wo	rking register to store the context data for a current connection that is being
3	processed.	
1	3.	The system of claim 1, wherein the cache is capable of storing and
2	retrieving co	entext data for multiple connections.
1	4.	The system of claim 1, wherein the interface comprises at least one of a
2	host interfac	e and a network interface.
1	5.	The system of claim 4, wherein the host interface interacts with a doorbell
2	queue, a con	appletion queue, and an exception/event queue.
1	6.	The system of claim 5, wherein each of the doorbell queue, the completion
2	queue, and tl	ne exception/event queue is a data structure.
1	7.	The system of claim 6, wherein each data structure has a priority
2	mechanism.	
1	8.	The system of claim 5, further comprising:
2	proce	essing logic to store the packet incoming from the host interface into the
3	doorbell que	ue;
4	host 1	nemory to store descriptors that are pointed to by the packet;
		37
		Dookst No. D17727

A network protocol processor system, the system comprising:

5	processing logic to access the descriptors in host memory for storage in the cache;	
6	and	
7	a scheduler to perform a hash based table lookup against the cache to correlate the	
8	packet with context data, to load the context into the working register when the context	
9	data is found in the cache, and to schedule a host memory lookup when the context data	
10	is not found in cache.	
1	9. The system of claim 8, further comprising:	
2	a Direct Memory Access (DMA) controller; and	
3	processing logic to notify the DMA controller to transfer data from host memory	
4	to the transfer queue.	
1	10. The system of claim 5, wherein the DMA controller is capable of storing	
2	data from the header and data queue into host memory.	
1	11. The system of claim 4, wherein the network interface interacts with a	
2	header and data queue and a transmit queue.	
1	12. The system of claim 11, further comprising:	
2	processing logic to store the packet incoming from the network interface into the	
3	header and data queue;	
4	a working register;	
5	a scheduler to perform a hash based table lookup against the cache to correlate the	
6	packet with context data, to load the context into the working register when the context	
7	data is found in the cache, and to schedule a host memory lookup when the context data	
8	is not found in cache.	
1	13. The system of claim 1, further comprising:	
2	a working register to store data for use by the processing engine; and	
3	a scheduler to locate and load the context data into the working register.	

1 14. The system of claim 1, further comprising: 2 a Direct Memory Access transfer queue; and 3 a Direct Memory Access receive queue. 1 15. The system of claim 1, further comprising: 2 a timer; and 3 a hardware assist to translate a virtual address to a physical address. 1 16 The system of claim 1, further comprising: 2 a thread cache to store intermediate system state; 3 a core receive queue; 4 a working register; 5 scratch registers; 6 a pieplined arithmetic logic unit; and 7 an instruction cache. 1 17. The system of claim 16, further comprising: 2 a high bandwidth connection between the thread cache and the working register 3 for parallel transfer of intermediate system state between the thread cache and the 4 working register. 1 18. The system of claim 1, further comprising: 2 an instruction cache to store code relevant to specific processing, while remaining 3 instructions are stored in at least one of host memory and cache to store context data. 1 19. The system of claim 1, further comprising a new instruction set including 2 context access instructions, hashing instructions, multi-threading instructions, Direct 3 Memory Access instructions, timer instructions, and network to host byte order 4 instructions.

1	20. The system of claim 1, further comprising:	
2	a scheduler coupled to the cache;	
3	a working register coupled to the cache; and	
4	processing logic in the processing engine to store context data in the working	
5	register into the storage area when processing of the packet has stalled.	
1	21. The system of claim 20, wherein the packet is a first packet and further	
	, and the second of the second	
2	comprising:	
3	processing logic to load context data for a second packet from the storage area	
4	into the working register.	
1	22. The system of claim 21, further comprising:	
2	processing logic to restore the context data for the packet into the working	
3	register.	
1	23. The system of claim 1, further comprising:	
2	a Direct Memory Access (DMA) controller coupled to the processing logic,	
3	wherein the DMA controller is capable of transferring data independently while the	
4	processing engine continues context processing in parallel.	
1	24. A network protocol processor system, the system comprising:	
2	a first interface to receive a first packet, wherein the first interface is coupled to a	
3	source of a first clock signal having a first frequency;	
4	a second interface to receive a second packet, wherein the second interface is	
5	coupled to a source of the first clock signal having the first frequency;	
6	processing logic to process the first packet and the second packet, wherein at least	
7	one component of the processing logic is coupled to a second clock signal having a	
8	second frequency different than the first frequency.	

1	25. The system of claim 24, wherein the second frequency is higher than the	
2	first frequency.	
1	26. The system of claim 24, further comprising:	
2	a computing device coupled to the first interface; and	
3	a network interface controller coupled to the second interface.	
1	27. The system of claim 24, further comprising:	
2	a storage area to store context data for at least one packet;	
3	a subset of the storage area to store context data for one packet; and	
4	processing logic to retrieve context data for the first packet, wherein the subset of	
5	the storage area is coupled to the processing logic.	
1	28. A method for processing a packet, comprising:	
2	receiving a packet;	
3	locating context data for the packet in a storage area; and	
4	processing the packet using the context data.	
1	29. The method of claim 28, further comprising:	
2	performing a lookup against the storage area to correlate the packet with context	
3	data;	
4	loading the context into a working register in response to locating the context data	
5	in the storage area; and	
6	scheduling a lookup of context data in response to determining that the context	
7	data is not in the storage area.	
1	30. The method of claim 29, further comprising:	
2	storing context data from the working register into the storage area when	
3	processing of the packet has stalled.	
_	processing of the packet has stanted.	

1	31. The method of claim 29, further comprising:	
2	updating process results to the working register;	
3	updating the storage area with the results in the working register; and	
4	updating a thread area with the results in the working register.	
1	32. The method of claim 31, wherein the packet is a first packet and further	
2	comprising:	
3	loading context data for a second packet from the storage area into the working	
4	register.	
1	33. The method of claim 32, further comprising:	
2	restoring the context data for the first packet into the working register.	
1	34. An article of manufacture comprising a storage medium having stored	
2	therein instructions that when executed by a computing device results in the following:	
3	receiving a packet;	
4	locating context data for the packet in a storage area; and	
5	processing the packet using the context data.	
1	35. The article of manufacture of claim 34, wherein the instructions when	
2	executed further result in the following:	
3	performing a lookup against the storage area to correlate the packet with contex	it.
4	data;	
5	loading the context into a working register in response to locating the context de	ata
6	in the storage area; and	
7	scheduling a lookup of context data in response to determining that the context	
8	data is not in the storage area.	
1	36. The article of manufacture of claim 35, wherein the instructions when	

2

executed further result in the following:

3	storing context data from the working register into the storage area when
4	processing of the packet has stalled.
1	37. The article of manufacture of claim 36, wherein the instructions when
2	executed further result in the following:
3	updating process results to the working register;
4	updating the storage area with the results in the working register; and
5	updating a thread area with the results in the working register.
1	38. The article of manufacture of claim 37, wherein the packet is a first packet
2	and wherein the instructions when executed further result in the following:
3	loading context data for a second packet from the storage area into the working
4	register.
l	39. The article of manufacture of claim 38, wherein the instructions when
2	executed further result in the following:
3	restoring the context data for the first packet into the working register.